

# On the Relationship Between Environmental and Mechanical Complexity in Evolved Robots

Joshua E. Auerbach<sup>1</sup> and Josh C. Bongard<sup>1</sup>

<sup>1</sup>Morphology, Evolution & Cognition Laboratory  
Department of Computer Science  
University of Vermont  
Burlington, VT 05405  
joshua.auerbach@uvm.edu

## Abstract

According to the principles of embodied cognition, intelligent behavior must arise out of the coupled dynamics of an agent's brain, body, and environment. This suggests that the morphological complexity of a robot should scale in relation to the complexity of its task environment. This idea is supported by recent work, which demonstrated that when evolving robot morphologies in simple and complex task environments more complex robot morphologies do tend to evolve in more complex task environments. Here this idea is extended to examining the mechanical complexity of evolved robots. Counter to intuition it is found that the mechanical complexity decreases in more complex task environments.

## Introduction

Proponents of embodied cognition posit that intelligent behavior is a product of the coupled dynamics between an agent's brain, body, and environment (Brooks, 1999; Anderson, 2003; Pfeifer and Bongard, 2006; Beer, 2008). Accordingly, the complexity of an agent's brain (control policy) as well as its physical body (morphology) should vary in proportion to the complexity of its task environment. Studying this hypothesis can be approached in several ways. One can investigate the relationship between control and morphology, as was done by Paul (Paul, 2006), and one can also study the relationship between task environment and morphology which is less well understood. In recent work (Auerbach and Bongard, 2012) we began to investigate this latter relationship by studying how the shape complexity of robot body parts varied when robots were evolved in more or less complex task environments. Here, that work is extended by studying a different aspect of morphological complexity: mechanical complexity, a function of the mechanical degrees of freedom of evolved robots.

The experiments presented in this paper fall within the domain of evolutionary robotics (ER) (Harvey et al., 1997; Nolfi and Floreano, 2000). In general ER refers to the practice of employing evolutionary algorithms for the purpose of creating robot control policies and/or morphologies. In the majority of ER studies, control strategies are evolved for human designed or bio-mimicked robot body plans, but it is

also possible to use evolutionary algorithms to create complete robots: placing not only robot control strategies under evolutionary control, but the robots' physical morphologies as well. Evolving morphology, in addition to control policy, allows for the discovery of body plans uniquely suited to a machine's given task environment and presents a systematic way to study the relationship between a robot's morphology and the task environment in which it evolved.

The idea of placing both the morphologies and controllers of robots acting in virtual environments under evolutionary control was first introduced by Sims (Sims, 1994). Sims' work has been followed by subsequent studies (e.g. Lund and Lee (1997); Adamatzky et al. (2000); Mautner and Belew (2000); Lipson and Pollack (2000); Hornby and Pollack (2001a); Komosinski and Rotaru-Varga (2002); Stanley and Miikkulainen (2003); Eggenberger (1997); Bongard and Pfeifer (2001); Bongard (2002); Auerbach and Bongard (2010a, 2011)) which also explored evolving the morphologies and control policies of simulated machines in virtual environments. These studies each had different methodologies and focuses, and the current work differs in a number of important ways.

The most visible ways in which the current study differs from all of these previous studies are (a) how morphological components are modeled and (b) the task environments within which robots evolve. In the majority of previous studies morphologies were built out of interconnected geometric primitives such as cuboids or spheres. These components are easy to model, but severely limit how complex an evolving morphology may become, and therefore restrict what task environments an evolved robot is able to succeed in. This was not a problem for the majority of earlier studies as they commonly restricted themselves to evolving locomotion over flat terrain: maximizing the distance that a robot can displace itself within a given amount of evaluation time. Here, however, more complex task environments are investigated that require the creation of more complex morphologies. Therefore, morphologies should be modeled in a manner which does not have such a low ceiling of complexity. Specifically, in the current work, morphologies are

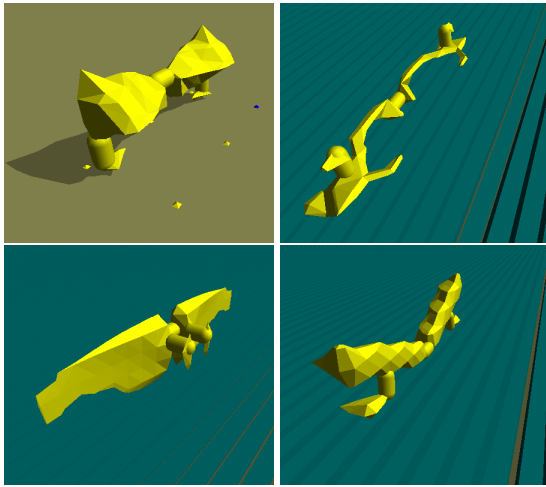


Figure 1: The simple flat ground control task environment (upper left) and three of the experimental task environments with robots that evolved to locomote in each. The ground is a high friction surface, while the blue “blocks of ice” have very low friction. Videos of these robots in action are available online at <http://tinyurl.com/ALife13-Videos>

composed of a number of triangular meshes (trimeshes). Trimeshes can model arbitrary shapes and thus allow for the creation of more complex morphologies than is possible with cuboids or spheres (see Figure 1 for examples).

The current study also differs from much previous work in this domain in the manner by which the robots’ genomes are encoded and evolved. Morphologies in the current work are encoded with Compositional Pattern Producing Network (CPPN) genomes (Stanley, 2007) which are evolved using CPPN-NEAT: an extension of the widely used NeuroEvolution of Augmenting Topologies (NEAT) algorithm (Stanley and Miikkulainen, 2001). CPPNs are a form of indirect encoding inspired by developmental biology possessing many advantages over other encodings (for more details see (Stanley, 2007; Stanley et al., 2009; Clune et al., 2009a,b; Auerbach and Bongard, 2010b,a, 2011)). This is particularly true for robot morphologies as it has been shown previously (Hornby and Pollack, 2001b; Komosinski and Rotaru-Varga, 2002) that generative and developmental encodings have demonstrable benefits over direct encodings in this domain.

Following the methods introduced in (Auerbach and Bongard, 2012), here robots are evolved not only to locomote over flat terrain, but to locomote in a number of more complex, icy task environments as well. However, while in that study robots were restricted to having two mechanical degrees of freedom, here robots are allowed more flexibility in their construction including the ability to utilize a greater number of degrees of freedom. How the robots evolve to

use (or not use) these additional degrees of freedom in different task environments is the main object of study. Here we define mechanical complexity to be the number of mechanical degrees of freedom in an evolved robot. This form of complexity can be considered an aspect of morphological complexity, but as will be shown, mechanical complexity is an orthogonal direction of complexity to the type of morphological complexity discussed in (Auerbach and Bongard, 2012), and provides additional insight into the relationship between task environments and the robots evolved inside them.

The rest of this paper is laid out as follows: first the CPPN encodings are described in more detail including how they evolve and how actuated robots are produced from them. Following this the simulated task environments in which robots are evolved are described including a brief discussion of previous experiments in these task environments and why the particular task environments employed here were chosen. Next, results are presented demonstrating how the mechanical complexity of evolved robots varies across these different task environments with counterintuitive results. This is followed by a discussion of these results and what conclusions may be drawn from them.

## Methods

### CPPNs

As mentioned in the introduction this study employs Compositional Pattern Producing Networks (CPPNs) for the purpose of encoding populations of evolving robots. CPPNs may be considered a form of artificial neural network (ANN). However, while traditional ANNs are often used as control policies for evolved robots, CPPNs are more often used as genomes for producing some other object of interest. Past work has employed CPPN genomes to evolve pictures (Stanley, 2007), 3D structures (Auerbach and Bongard, 2010b; Clune and Lipson, 2011), robot morphologies (Auerbach and Bongard, 2010a, 2011) or traditional ANNs themselves (Stanley et al., 2009; Clune et al., 2009a; Verbancsics and Stanley, 2011). Here CPPNs are similarly employed to produce actuated robot morphologies.

CPPNs differ from traditional ANNs in several other important ways. While traditional ANNs typically use the same activation function (such as a sigmoid or a step function) at every node, CPPN nodes can take on one of several activation functions from a predefined set. This set typically contains functions that are symmetric such as Gaussian as well as repetitive functions such as sine or cosine. Using functions with these properties allows CPPNs to produce outputs with properties commonly seen in natural systems: symmetry, repetition, and repetition with variation. A more thorough discussion of CPPNs and their properties is beyond the scope of this paper. More details are available elsewhere in the literature (Stanley (2007) for example).

## Evolutionary Algorithm

Similar to most other studies employing CPPN genomes, the CPPN-NEAT (Stanley, 2007) evolutionary algorithm is employed to evolve CPPNs in this work. In CPPN-NEAT the state of the art NeuroEvolution of Augmenting Topologies (NEAT) (Stanley and Miikkulainen, 2001) algorithm for neuro-evolution is extended to evolve CPPNs. In this algorithm the CPPNs in the initial population are created to be minimally complex. That is, initially the networks do not have any internal or hidden nodes. Over evolutionary time the complexity of networks in the population is allowed to gradually increase through the creation of additional nodes and links. Often adding additional components to an evolving network will cause the fitness of its phenotype to decrease. NEAT compensates for this by dividing the population into “species” thus allowing novel structural innovations time to mature and promoting genotypic diversity to prevent pre-mature convergence to local optima. For a complete description of how NEAT and CPPN-NEAT work, and further discussion of their beneficial properties, the reader is directed to (Stanley and Miikkulainen, 2001; Stanley, 2007).

## Building Robots from CPPNs

Recently (Auerbach and Bongard, 2012) we introduced a system for using CPPNs to create actuated robot morphologies composed of triangular mesh components, which is extended here. This method differs from previous studies (Auerbach and Bongard, 2010a, 2011) where robots were constructed from evolving CPPNs by attaching spherical components to each other by means of an iterated growth procedure. While these earlier studies produced promising results, the methods they employed have several undesirable properties. The extra indirection created by the growth procedure used there prevents many of the desirable features of CPPNs (discussed above) from being realized in the morphologies they produce. Additionally, while it is easy to physically simulate spheres as they have single points of contact, it is possible to create much more complex morphologies using trimeshes.

Trimeshes do require more computational resources to simulate however, as they do not have such simple contact models as spheres, and require the use of smaller simulation step sizes to be stable in the task environments investigated here. However, all experiments described in this paper were carried out on a 7.1 teraflop supercomputing cluster<sup>1</sup>, thus making these simulations feasible.

As opposed to employing a growth procedure to create morphologies from CPPNs the current study employs a voxel based method to create morphologies out of trimesh components. This is similar to what is done for the creation of 3D shapes in (Clune and Lipson, 2011). A regular grid is

placed over a region of 3D-space which defines the presence of voxel locations. In the current work this region extends from  $-1$  to  $1$  (inclusive) in each dimension and grid lines are placed at intervals of  $0.2$ . This yields a total of  $11$  grid lines in each dimension for a total of  $1331$  voxels, this is the same discretization that was applied in (Auerbach and Bongard, 2012).

A candidate CPPN is iteratively queried with the  $(x, y, z)$  Cartesian coordinates at every voxel location except for the extrema in each direction. Voxel locations that exceed a pre-defined output threshold ( $0.5$  in this case) are considered to contain matter, while those that do not exceed this threshold are considered to be devoid of matter. All voxels lying on one of the extrema ( $|x| = 1$  or  $|y| = 1$  or  $|z| = 1$ ) are given output value  $0$  to ensure that the final triangular meshes have completely enclosed surfaces. Once the CPPN has been queried for every voxel location, the Marching Cubes algorithm (Lorensen and Cline, 1987) is employed to create triangular meshes from the underlying voxel data. Specifically an enclosed triangular mesh is created for each connected voxel component which defines the exterior surface of a single physical shape. These triangular meshes are sent to the physics simulator where they define the exterior surfaces of solid objects and are imbued with mass. As far as the authors are aware prior to (Auerbach and Bongard, 2012) physically simulating evolved, rigid body robots composed of triangular meshes had not been previously reported in the literature.

Our previous work concerned itself with investigating how different task environments affect the shapes of evolved morphologies. To accomplish this goal a single enclosed trimesh component out of the many possibly produced from a CPPN was selected and then reflected and copied in order to form a bilaterally symmetric, two mechanical degree of freedom, actuated robot. Here, however, the primary object of study is the mechanical complexity of the evolved robots, so more components are needed. The current system requires that a candidate CPPN produce at least two enclosed trimesh components. The two largest components  $A$  and  $B$  are then selected to produce an actuated robot. This is done as follows. First the vertices  $a \in V(A)$  and  $b \in V(B)$  are found that minimize

$$|\vec{ab}| \quad \forall (a, b) \in V(A) \times V(B)$$

where  $V(A), V(B)$  are the vertices of  $A, B$  respectively. Next, the component with larger minimum  $z$ -coordinate of  $A, B$  is translated along  $\vec{ab}$  (or  $\vec{ba}$ ) until it is  $0.2$  units away from the other component, and the two components are connected together via an intermediary capsule (capped cylinder) of length  $0.2$  units and radius  $0.1$  with major axis defined by  $\vec{ab}$ . The trimesh components may connect via this intermediary capsule by means of two joints, each being a single degree of freedom rotational (hinge) joint. These joint will have rotation normals determined by  $\vec{ab}$ . Specifically

<sup>1</sup>The Vermont Advanced Computing Core (VACC),  
<http://www.uvm.edu/vacc>

Parameter Name	Symbol	Range of allowed values	Interpretation
<b>Enable Flag</b>	$f$	[0.0, 1.0]	If $f > 0.5$ , then the corresponding joint is enabled, else disabled.
<b>Amplitude</b>	$a$	[0.25, 0.75]	If the joint is enabled, then it is actuated by an oscillation between $-a\pi$ and $a\pi$ radians, additionally the joints range of motion is restricted to this range.
<b>Period</b>	$p$	[250.0, 1500.0]	If the joint is enabled, then its oscillation will have a period of $p$ simulation time steps
<b>Phase Shift</b>	$s$	[-1.0, 1.0]	If the joint is enabled, then its oscillation will be offset from the global oscillation by $s$ periods

Table 1: Description of the four floating point parameters evolved for each of the six potential mechanical degrees of freedom.

two rotation normals that are orthogonal to each other and orthogonal to  $\vec{ab}$  are chosen. Since these two joints effectively define a universal joint, the specific normals are unimportant as long as they are orthogonal to each other and to  $\vec{ab}$ , so the first  $\vec{n}_1$  is chosen arbitrarily (but consistently) to be orthogonal to  $\vec{ab}$  and the second  $\vec{n}_2$  is computed as  $\vec{ab} \times \vec{n}_1$ .

Once the two trimesh components are connected together with their intermediary capsule the whole object including the connecting joints is reflected across the  $x$ -axis as was done with the single trimesh component in (Auerbach and Bongard, 2012). These objects are then spread apart by 0.2 units and once again connected by a capsule of this length. This capsule has its major axis along the  $y$ -axis of the coordinate system and connects the two objects at their closest points. These objects each connect to this capsule by means of hinge joints. These joints have rotation normals of (1, 0, 0) and (0, 0, -1) such that the joints rotate through the robot’s coronal and sagittal planes respectively. Reflecting and copying the object in this manner ensures that the robots are bilaterally symmetric, which makes locomotion easier, while using two evolved trimesh components instead of the one used in prior work allows for a much greater number of morphologies and locomotion strategies. The two components within each half of the robot may connect in any orientation, and the robots may now have up to six mechanical degrees of freedom.

In addition to the trimesh producing CPPNs, each robot genome possesses a number of additional parameters that are directly encoded as was done in (Auerbach and Bongard, 2012). These parameters are stored as floating point values and are used to determine aspects of the control policy as well as mechanical properties of the evolving robots. Principally, there are six parameters, one for each potential mechanical degree of freedom that act as flags for enabling or disabling a given joint. If a joint is disabled it is replaced with a rigid connection and the remainder of the control parameters relating to that joint are ignored. However, if a joint is enabled it is actuated by means of a coupled oscillator parameterized by its amplitude, period, and phase shift from a global sinusoidal pattern generator. This results in the complete genomes being composed of a CPPN plus a 24-dimensional floating point array (four parameters for each of

the six potential degrees of freedom). These floating point values are recombined and mutated in the same manner as CPPN link weights with mutation magnitudes scaled by the range of values for that parameter. Additionally, crossover on these vectors is possible in all instances of sexual reproduction since every individual contains a vector of the same dimensionality. These parameters, their ranges, and their meanings are detailed in Table 1. Each parameter has a mutation probability of 0.1, same as used in (Auerbach and Bongard, 2012).

Allowing each degree of freedom to be enabled or disabled in this manner allows evolution to adjust the number of mechanical degrees of freedom as necessary and therefore be able to tune the mechanical complexity of the evolved robots. Moreover, encoding the control parameters in this fashion is done to keep the controllers as simple as possible so that fitness is primarily dictated by the morphologies of the robots while at the same time allowing for diverse enough behavior so that the robots can succeed in the different task environments investigated.

### Selecting desirable robots

A candidate robot, including two enclosed triangular meshes, joint enable flags, and accompanying control parameters are sent to a physics simulator<sup>2</sup> and allowed to act for a fixed number of simulation time steps. Similar to (Auerbach and Bongard, 2012) robots are allowed to move for  $T = 12500$  time steps. While this is a much greater number of time steps than has been employed in earlier studies (e.g. 2500 in (Auerbach and Bongard, 2011)) it is chosen in order to simulate a comparable amount of real world time. The reason such a large  $T$  is necessary is because a very small step size of 0.001s is used in this work. This small step size is necessary to stably simulate the sorts of simulated robots employed here in complex environments.

After the robot has completed its time in the simulator its fitness is calculated. This fitness calculation is exactly the same used in (Auerbach and Bongard, 2012). It is designed to prevent evolution from “cheating” as it often does

<sup>2</sup>Simulations are conducted in the Open Dynamics Engine (<http://www.ode.org>), a widely used open source, physically realistic, simulation environment.

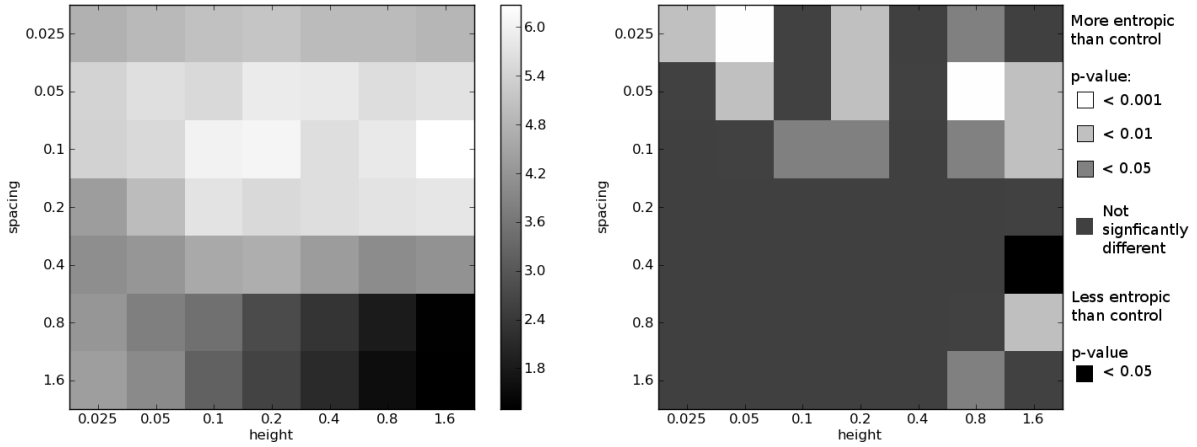


Figure 2: Results from (Auerbach and Bongard, 2012). **(Left)** Mean distance achieved (in arbitrary ODE units) by best individual in final generation taken across 100 independent runs in each of 49 experimental task environments investigated there. For comparison the mean distance achieved from 100 independent runs in a flat ground control task environment was 5.09 units. **(Right)** The ways in which morphologies from experimental environments were more, less, or equally complex (entropic) compared to those evolved in the control task environment. The more complex experimental task environments tended to select for more complex morphologies: there were many experimental task environments where significantly more complex morphologies evolved, while only one experimental task environment selected for significantly less complex morphologies. All p-values were calculated using the Mann-Whitney U test. Figure taken from (Auerbach and Bongard, 2012)

with naïve fitness functions. While a detailed explanation of the ways in which evolution may “cheat” different fitness functions is provided in that paper, here we simply state that fitness is calculated as  $\min p(T)_x - \max p(0)_x$  where  $\min p(T)_x$  is the minimum  $x$ -coordinate of any point on the robot at time  $T$ , and  $\max p(0)_x$  is the maximum  $x$ -coordinate of any point on the robot at the start of the evaluation.

Using this method of fitness evaluation robots are evolved with CPPN-NEAT for 500 generations with a population size of 150 individuals. The implementation of CPPN-NEAT including its parameter settings and CPPN activations functions are the same as employed in (Auerbach and Bongard, 2012).

### Choosing task environments

Previously, with the robots composed of a single enclosed trimesh that was reflected and copied, we explored evolving robots in a large number of task environments with the goal of studying how morphological complexity varies in relation to environmental complexity. These task environments consisted of a control environment with flat, high friction ground similar to that used in many other studies, and experimental task environments with an infinite series of low frictions rectangular solids, or “blocks of ice”, fixed in place on top of the ground. These “ice blocks” were constructed such that it was impossible for a robot to gain purchase by moving over their upper surfaces but needed instead to reach into the gaps between the blocks to propel themselves forward. This required the evolution of morphologies with appropri-

ate physical forms. A large number of these icy task environments were explored varying according to two parameters: the height of the blocks and the spacing between the blocks. While the relative complexities of different icy environments were not considered, all the icy environments are considered to be more complex than flat ground because they have greater Kolmogorov Complexity (Kolmogorov, 1965).

Figure 2 revisits these results. It shows, for robots evolved in that work, both how mean fitness varied across task environments and how the evolved robot morphologies differed in complexity when compared to those evolved to locomote in the flat ground, control, environment<sup>3</sup>. These results are employed here to select task environments for investigation with the current system.

Robots evolved with the current system, employing two trimesh components and three capped cylinders with up to six actuated mechanical degrees of freedom, are slower to simulate than those evolved previously. Due to this slowness, and additional time constraints, it was not possible to experiment with evolving robots in all 50 task environments previously investigated. In lieu of that, robots in the current study are evolved in the flat ground control environment plus five experimental environments. These five experimental environments are chosen based on previous results to be those

<sup>3</sup>The measure used for comparing morphological complexities,  $H_\Delta$ , is a measure of shape complexity based on Shannon Entropy (Shannon, 1948) that has been previously shown to correlate with human intuitions of complexity (Page et al., 2003; Sukumar et al., 2008). The reader is referred to (Auerbach and Bongard, 2012) for a description of this measure.

within which robots could be successful and which selected for the most morphologically complex robots (see Figure 2). Specifically the five environments chosen are: blocks of ice 0.8 units tall spaced by 0.05 units (*Environment 1*), blocks of ice 0.05 units tall spaced by 0.025 units (*Environment 2*), blocks of ice 1.6 units tall spaced by 0.1 units (*Environment 3*), blocks of ice 1.6 units tall spaced by 0.05 units (*Environment 4*), and blocks of ice 0.2 units tall spaced by 0.05 units (*Environment 5*). These five task environments cover a variety of these parameters and should be a good sampling of the overall parameter space.

## Results

For each of the six task environments investigated: the control plus five experimental task environments, 50 independent experimental runs of CPPN-NEAT were conducted<sup>4</sup>. As can be seen in Figure 3, in each environment studied this system is capable of evolving robots that successfully locomote in the desired direction. Though, due to using the same number of evaluations in an enlarged search space the robots produced in the final generations here tend not to locomote as far as those evolved previously (compare to the left of Figure 2). However, the absolute performance of these robots is not of primary interest in this paper.

Of greater concern is how the mechanical complexity of the evolved robots varies from the simple control environment to the more complex experimental task environments. Towards this aim Figure 4 plots the mean number of mechanical degrees of freedom that robots evolved to use in each task environment. Counter to intuition the simple task environment actually selects for more mechanically complex robots: the robots evolved in the simple task environment have significantly more mechanical degrees of freedom on average, than those evolved in each of the five complex task environments. This is corroborated by Figure 5 which shows that the flat ground task environment not only selects for a greater number of mechanical degrees of freedom but that the degrees of freedom that are selected for have a significantly greater range of motion on average than the degrees of freedom in robots evolved in each of the more complex experimental task environments.

## Discussion

Why is it that the same task environments which have been shown to select for greater complexity of morphological components select for reduced mechanical complexity? Intuitively these two forms of complexity should be correlated, but this is clearly not the case here. One hypothesis is that the reduction of mechanical complexity in the icy task environments is due to them being more difficult than the flat

<sup>4</sup>While 50 runs were started for each task environment, a small number of runs failed to complete for each of the experimental task environments. The results reported here only include those runs that completed successfully.

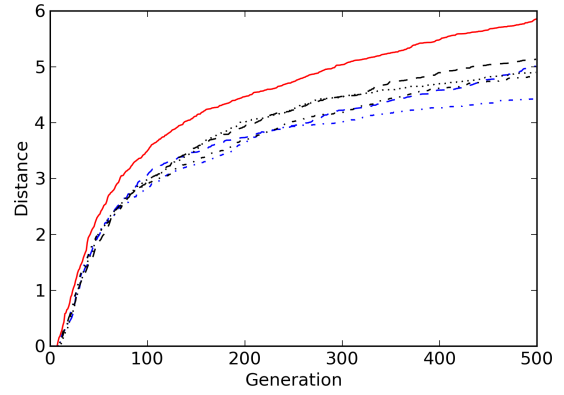


Figure 3: Mean distances by generation achieved by robots evolved in the control environment (red) and each of the five experimental task environments (*env. 1* blue dashes, *env. 2* blue dash-dots, *env. 3* black dashes, *env. 4* black dash-dots, *env. 5* black dots).

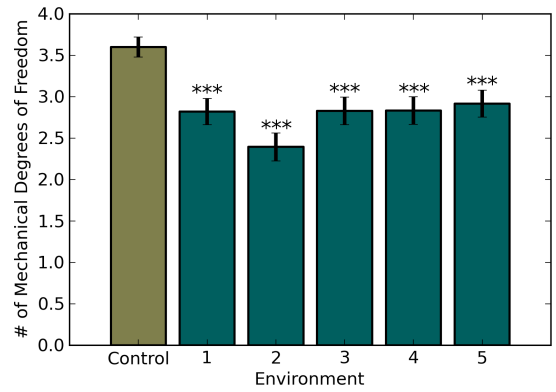


Figure 4: Mean number of mechanical degrees of freedom with standard errors for robots evolved in each task environment. Robots evolved in each of the icy task environments have significantly fewer mechanical degrees of freedom than those evolved in the control environment,  $p$ -values  $< 0.001$  in all cases (Mann-Whitney U test).

ground task environment. As can be seen in Figure 3 robots are not able to evolve to locomote as far in the icy task environments as they are on flat ground. This suggests there may be fewer ways to succeed in the icy task environments, and if it is easier to succeed with less mechanical complexity than there will be selection pressure in that direction. Meanwhile, if flat ground is an easier task environment regardless of mechanical complexity there will be little selection pressure on the number of degrees of freedom of the robots evolved there. However, if this is the case, one would expect each degree of freedom of robots evolved on flat ground to be enabled or disabled with equal probability. But, from looking at Figure 4 it can be seen that this is clearly not the case. Robots evolved in the flat ground task environment



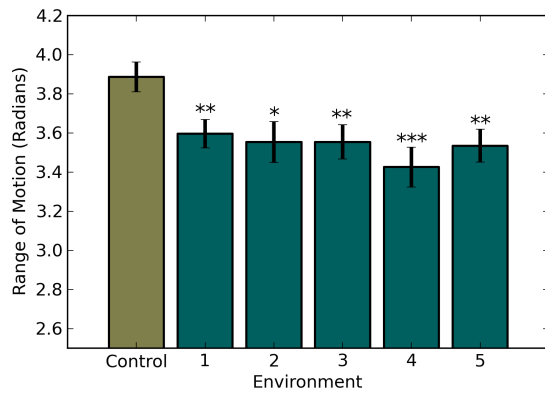


Figure 5: Mean range of motion in radians taken across each enabled joint (mechanical degree of freedom) with standard errors. Robots evolved in each of the icy task environments have significantly smaller ranges of motion than those evolved in the control environment. \* denotes  $p$ -values  $< 0.05$ , \*\* denotes  $p$ -values  $< 0.01$ , and \*\*\* denotes  $p$ -values  $< 0.001$  (Mann-Whitney U test).

have a significantly greater number of degrees of freedom than the three that would be expected by equal probability.

Another hypothesis is that there is simply an advantage to having less mechanical complexity in the icy task environments investigated. Succeeding in these environments involves reaching into the gaps between blocks in order to gain purchase, and then coming out of the gaps in order to move forward. Since the robots evolved in this work are all driven by open loop controllers, they have no way of sensing when they are in the gaps or not. It may be that extra mechanical degrees of freedom make it more difficult for the robot to get out of its own way as it traverses the environment. In other words extra mechanical degrees of freedom driven by a sinusoidal control signal cause the robot to often catch itself in the gaps when it could be gliding forward. This seems likely to be the case. As can be seen in the video available at <http://tinyurl.com/alifel3-1DOF> it is possible for robots to succeed in these task environments with only a single mechanical degree of freedom and the proper physical shape. This robot only has one actuated joint rotating horizontally but due to its shape it is able to fall into the gaps, gain purchase and glide out of them. Several such single degree of freedom robots evolved in the icy task environments, but only one such robot evolved in the control task environment and it has substantially lower fitness.

While it is counter-intuitive that task environments that select for more complex body components select for less mechanical complexity it makes sense in this instance. It is likely, however that other task environments that are complex in different ways will select for robots that have complex body components and are more mechanically complex. For instance if there existed other obstacles in the environ-

ment that the robot needs to step over one could imagine how additional degrees of freedom would be useful in order to reach over the obstacles in order to gain purchase on their far sides in ways that would not be possible without additional degrees of freedom. Likewise if the spacing between blocks was uneven then most likely the open loop control policies employed here would be unable to succeed. If sensors and closed loop control were employed it may be advantageous to have extra degrees of freedom in order to actively sense the environment and decide how to move.

## Conclusion

This work has investigated the relationship between environmental and mechanical complexity in evolved robots. Results of previous work were used to select task environments in which successful, morphologically complex, robots were previously evolved. However, counter to intuition, the robots evolved here were less mechanically complex than those evolved in a simpler control task environment. This demonstrates that these different forms of morphological complexity do not necessarily correlate with each other, but are likely orthogonal.

Moving forward it will be interesting to explore evolving robots in other task environments that are complex in different ways. It is likely that while the task environments investigated here do not select for greater mechanical complexity there exist task environments in which both greater mechanical complexity and greater complexity of body shape will be selected for. Additionally it will be of interest how control complexity varies in relation to these morphological complexity measures. To this aim the current evolutionary system will be extended to allow for more sophisticated closed loop neural network controllers. Are the task environments that select for greater morphological complexity in one way or another also those that select for greater control complexity? Or are these different forms of complexity—morphological, mechanical, and control—independent?

## Acknowledgements

This work was supported by National Science Foundation Grant PECASE-0953837 and DARPA M3 grant W911NF-1-11-0076.

The authors also acknowledge the Vermont Advanced Computing Core which is supported by NASA (NNX 06AC88G), at the University of Vermont for providing High Performance Computing resources that have contributed to the research results reported within this paper.

## References

- Adamatzky, A., Komosinski, M., and Ulatowski, S. (2000). Software review: Framsticks. *Kybernetes: The International Journal of Systems & Cybernetics*, 29(9/10):1344–1351.
- Anderson, M. (2003). Embodied Cognition: A field guide. *Artificial Intelligence*, 149(1):91–130.

- Auerbach, J. E. and Bongard, J. C. (2010a). Dynamic Resolution in the Co-Evolution of Morphology and Control. In *Artificial Life XII: Proceedings of the Twelfth International Conference on the Simulation and Synthesis of Living Systems*.
- Auerbach, J. E. and Bongard, J. C. (2010b). Evolving CPPNs to Grow Three-Dimensional Physical Structures. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*.
- Auerbach, J. E. and Bongard, J. C. (2011). Evolving Complete Robots with CPPN-NEAT: The Utility of Recurrent Connections. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*.
- Auerbach, J. E. and Bongard, J. C. (2012). On the Relationship Between Environmental and Morphological Complexity in Evolved Robots. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*. To Appear.
- Beer, R. D. (2008). The dynamics of brain-body-environment systems: A status report. In Calvo, P. and Gomila, A., editors, *Handbook of Cognitive Science: An Embodied Approach*, pages 99–120. Elsevier.
- Bongard, J. and Pfeifer, R. (2001). Repeated structure and dissociation of genotypic and phenotypic complexity in Artificial Ontogeny. *Proceedings of The Genetic and Evolutionary Computation Conference (GECCO 2001)*, pages 829–836.
- Bongard, J. C. (2002). Evolving modular genetic regulatory networks. In *Proceedings of The IEEE 2002 Congress on Evolutionary Computation (CEC2002)*, pages 1872–1877.
- Brooks, R. (1999). *Cambrian intelligence*. MIT Press Cambridge, Mass.
- Clune, J., Beckmann, B., Ofria, C., and Pennock, R. (2009a). Evolving Coordinated Quadruped Gaits with the HyperNEAT Generative Encoding. In *Proceedings of the IEEE Congress on Evolutionary Computing*, pages 2764–2771.
- Clune, J. and Lipson, H. (2011). Evolving 3d objects with a generative encoding inspired by developmental biology. In *Proceedings of the Eleventh European Conference on Artificial Life (ECAL)*, pages 144–148.
- Clune, J., Pennock, R. T., and Ofria, C. (2009b). The sensitivity of hyperneat to different geometric representations of a problem. In *Proceedings of the Genetic and Evolutionary Computation Conference*.
- Eggenberger, P. (1997). Evolving morphologies of simulated 3D organisms based on differential gene expression. *Procs. of the Fourth European Conf. on Artificial Life*, pages 205–213.
- Harvey, I., Husbands, P., Cliff, D., Thompson, A., and Jakobi, N. (1997). Evolutionary robotics: the sussex approach. *Robotics and Autonomous Systems*, 20:205–224.
- Hornby, G. and Pollack, J. (2001a). Body-brain co-evolution using l-systems as a generative encoding. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, pages 868–875.
- Hornby, G. and Pollack, J. (2001b). Evolving L-systems to generate virtual creatures. *Computers & Graphics*, 25(6):1041–1048.
- Kolmogorov, A. N. (1965). Three approaches to the quantitative definition of information. *Problems of Information Transmission*, 1(1):1–7.
- Komosinski, M. and Rotaru-Varga, A. (2002). Comparison of different genotype encodings for simulated three-dimensional agents. *Artif. Life*, 7(4):395–418.
- Lipson, H. and Pollack, J. B. (2000). Automatic design and manufacture of artificial lifeforms. *Nature*, 406:974–978.
- Lorensen, W. E. and Cline, H. E. (1987). Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH Comput. Graph.*, 21:163–169.
- Lund, H. H. and Lee, J. W. P. (1997). Evolving robot morphology. *IEEE International Conference on Evolutionary Computation*, pages 197–202.
- Mautner, C. and Belew, R. (2000). Evolving robot morphology and control. *Artificial Life and Robotics*, 4(3):130–136.
- Nolfi, S. and Floreano, D. (2000). *Evolutionary Robotics: The Biology, Intelligence, and Technology*. MIT Press, Cambridge, MA, USA.
- Page, D., Koschan, A., Sukumar, S., Roui-Abidi, B., and Abidi, M. (2003). Shape analysis algorithm based on information theory. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, volume 1, pages I – 229–32 vol.1.
- Paul, C. (2006). Morphological computation: A basis for the analysis of morphology and control requirements. *Robotics and Autonomous Systems*, 54(8):619–630.
- Pfeifer, R. and Bongard, J. (2006). *How the Body Shapes the Way We Think: A New View of Intelligence*. MIT Press.
- Shannon, C. E. (1948). A mathematical theory of communication. *Bell system technical journal*, 27.
- Sims, K. (1994). Evolving 3D morphology and behaviour by competition. *Artificial Life IV*, pages 28–39.
- Stanley, K., D’Ambrosio, D., and Gauci, J. (2009). A Hypercube-Based encoding for evolving Large-Scale neural networks. *Artificial Life*, 15(2):185–212.
- Stanley, K. and Miikkulainen, R. (2003). A taxonomy for artificial embryogeny. *Artificial Life*, 9(2):93–130.
- Stanley, K. O. (2007). Compositional pattern producing networks: A novel abstraction of development. *Genetic Programming and Evolvable Machines*, 8(2):131–162.
- Stanley, K. O. and Miikkulainen, R. (2001). Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10:2002.
- Sukumar, S., Page, D., Koschan, A., and Abidi, M. (2008). Towards understanding what makes 3d objects appear simple or complex. In *IEEE Conference on Computer Vision and Pattern Recognition CVPR 2008, Sixth IEEE Workshop on Perceptual Organization in Computer Vision (POCV)*.
- Verbancsics, P. and Stanley, K. O. (2011). Constraining Connectivity to Encourage Modularity in HyperNEAT. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*.